An Experimental Pipeline for Automated Reasoning in Natural Language

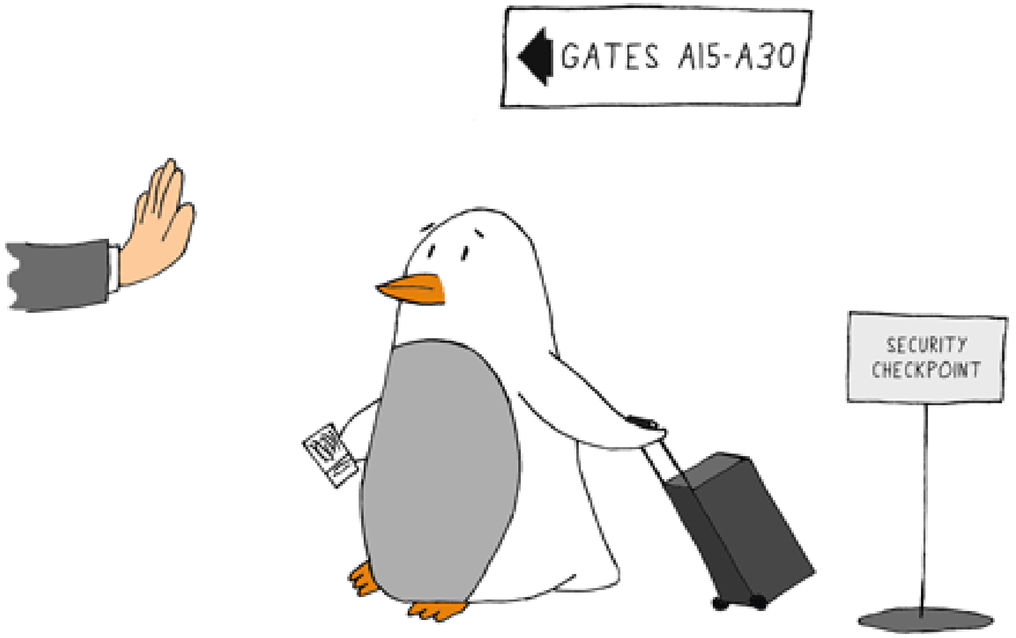
Tanel Tammet, Dirk Draheim, Priit Järv, Martin Verrev

Tallinn University of Technology, Estonia

A small example of NL input / output

Birds can fly. Penguins are birds. Penguins cannot fly.
Patrick is a penguin and Billie is a bird.
Mickey is perhaps a bird. Theo is possibly a penguin.

Who can fly?
Billie and maybe Mickey.
Who cannot fly?
Patrick and likely Theo.
Patrick can fly?
No.
Billie can swim?
Unknown.



"I'm sorry sir, but you're on the no-fly list."
WWW.ZOITZ.COM

Some example tests from AllenNLP

Given

The circuit has / does not have a switch. The circuit has a bell. The switch is on/off. If the circuit has the switch and the switch is on then the circuit is complete. If the circuit does not have the switch then the circuit is complete. If the circuit is complete and the circuit has the light bulb then the light bulb is glowing. If the circuit is complete and the circuit has the bell then the bell is ringing. If the circuit is complete and the circuit has the radio then the radio is playing.

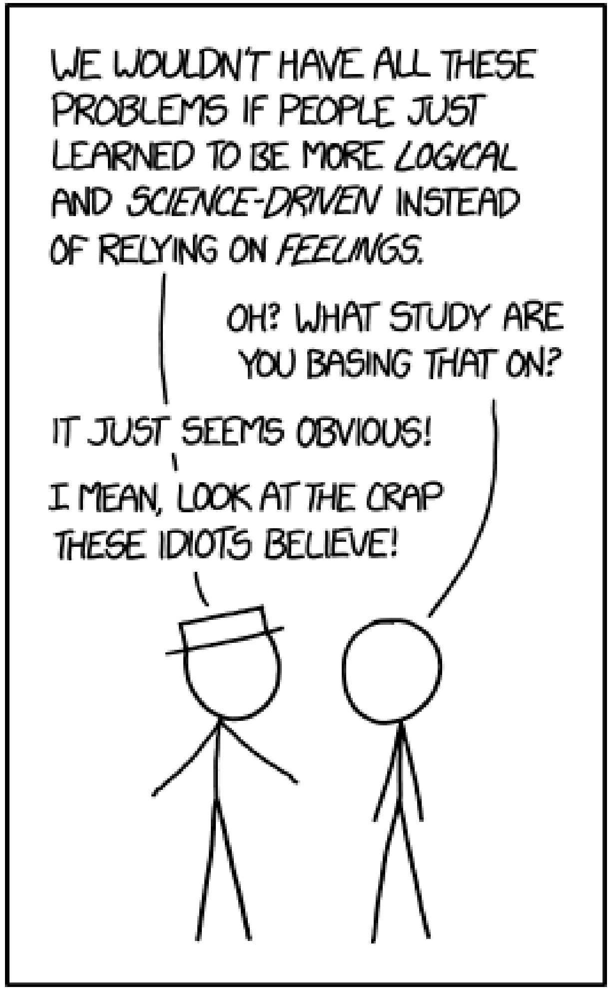
Ask

The bell is ringing? The radio is playing? The light bulb is glowing?

Background

Question answering and inference using NL

- Strongly related to commonsense reasoning.
- Long history, little success with symbolic methods. CYC!?
- Large Language Model (LLM) explosion: success with issues
    - Explainability
    - Hallucinations
    - Hard to connect with tools like databases, computation, etc
    - Hard to do deep inference
    - Controllability: how to add a lot of rules?
    - etc



Tiny GPT examples

- Finds the correct answer :

  If an animal likes honey, then it is probably a bear. Most bears are big, although young bears are not big. John is an animal who likes honey. Mike is a young bear. Who is big?

- Does not find the correct answer:

  If an animal likes honey, then it is probably a bear. Most bears are big, although young bears are not big. John is an animal who likes honey. Mike is a young bear. Mike can eat a lot. Penguins are birds who cannot fly. John took the block from the colored table. The table was really nice. The robot arm lifted a blue block from the table. Who is big?

Why is symbolic reasoning with NL hard?

- Semantic parsing, i.e. translating natural language to logic, is extremely hard due to the highly complex and exception-rich nature of natural language.
- Existing knowledge bases of "common sense" do not cover a critical mass of the basic understanding of the world even a small child possesses.
- Classical first order reasoning itself cannot cope with contradictory knowledge items, probabilistic or uncertain information and exceptions to rules.
- Finding logic-based proofs often requires long derivations and the huge knowledge base causes a quick combinatorial explosion of the search space.

Hypothesis: ML and LLMs can alleviate all the main issues above

Neurosymbolic / hybrid systems

- A new area, much smaller than end-to-end learning
- Most of the research investigates ways to use rules while learning
- Our direction: start with the symbolic reasoner and find ways to add learning and integrations with LLM systems

- What we have: a symbolic pipeline for commonsense reasoning with NL, geared towards experimenting with ML and LLM augmentation

Pipeline details

Main parts

http://github.com/tammet/nlpsolver

Input text + question

→ using the semantic parser →

FOL + confidences and exceptions

→ adding background knowledge (world model) →

FOL input to the reasoner

→ using the commonsense reasoner GK →

Answers, confidences, proofs in json

→ using the proof-to-English converter →

Explanations in English

Semantic parsing

- Takes English strings of natural language text as input and outputs clausified first-order logic formulas encoded in JSON.
- The main extension to FOL is
  - adding numerical confidence to clauses
  - implementing default logic (defeasible reasoning)
- Our parser consists of a number of phases, each adding new structural details to the results of the previous phases.
- Implemented procedurally, i.e. no conversion rules.

Main phases of parsing

- A few iterations of:
  - Textual simplification rules
  - NL to Universal Dependencies (UD) using the Stanford Stanza parser.
  - Simplification rules from UD patterns back to NL
- Convert UD representation to logic.
- Use a specialized clausification procedure (for example, spread the confidence number over suitable parts).
- Simplify.

Subphase: convert UD representation to logic

- Restructure the UD graph to a semi-logical representation explicating the outward logical structure around the subject/verb, object/verb or subject/verb/object tuples.
- Convert this representation to quantified logic, using the details of the original UD structure to find additional critical information like articles, negation, different kinds of quantifiers etc.
- Some of the hard tasks: coreference resolution (The wolf saw a rabbit. It was hungry.) and determining whether a noun stands for a concrete object or should be quantified over (Animals run to the lake. Bears like honey.).
- Add confidence numbers and exceptions (Most bears are big).
- Generate generic instances (Birds can fly. Who can fly?)

Example

John is a nice animal who likes honey.

isa(animal, c1_John)

prop(nice, c1_John, generic, generic, ctxt(Pres, 1))

def0(c1_John)

forall S : (def0(c1_John) ⇔

exists X : (isa(honey, X) & exists A : do2(like, c1_John, X, A, ctxt(Pres, S)))))

Example

Most bears are big.

0.85: forall X : isa(bear, X) =>

prop(big, X, generic, generic, ctxt(Pres, 1))) |

block(h(bear,1), neg(prop(big, X, generic, generic, ctxt(Pres, 1))))

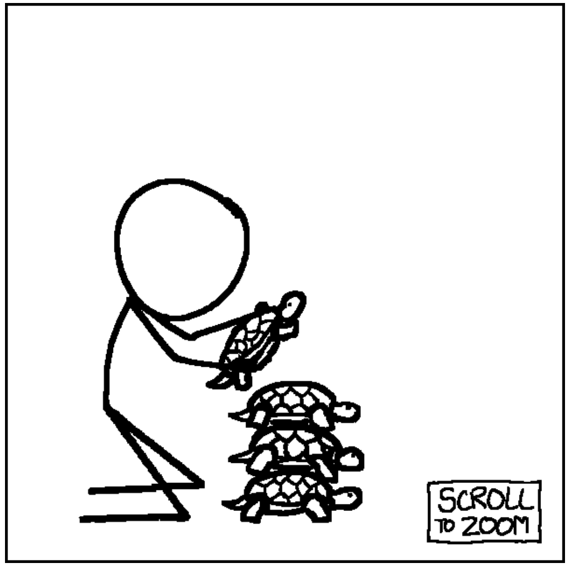
Typical regression test examples for parsing

Given
A blue hand of a man moved a wheel of a large wheelbarrow.
Ask
A hand moved a wheel of a small wheelbarrow? -> Unknown
The man had a hand? -> True
The man had a blue hand? -> True
The man had a red hand? -> Unknown
The man had a wheel? -> Unknown
The wheelbarrow had a wheel? -> True
A large wheelbarrow had the wheel? -> True
A small wheelbarrow had the wheel? -> Unknown
The large wheelbarrow had a wheel? -> True
The small wheelbarrow had a wheel? -> Unknown
The wheelbarrow had a hand? -> Unknown

Integration with knowledge bases

- To answer the query “Tweety is a bird. Can Tweety fly?”, the system needs to have the background knowledge that birds can fly.
- The knowledge base provides the world model of our reasoning system.
- We construct the knowledge base (KB) using default logic rules augmented with numeric confidences.
- A small part of the knowledge base forms a core world model and is built by hand.
- The bulk of the knowledge is integrated automatically from existing common sense knowledge bases of triplets : ConceptNet, WebChild, Aristo TupleKB, Quasimodo, Ascent++, UnCommonSense, ATOMIC.
- We have built a specialized pattern matching semantic parser to convert the relations to first order logic rules with the default logic extensions and estimated numeric confidence.

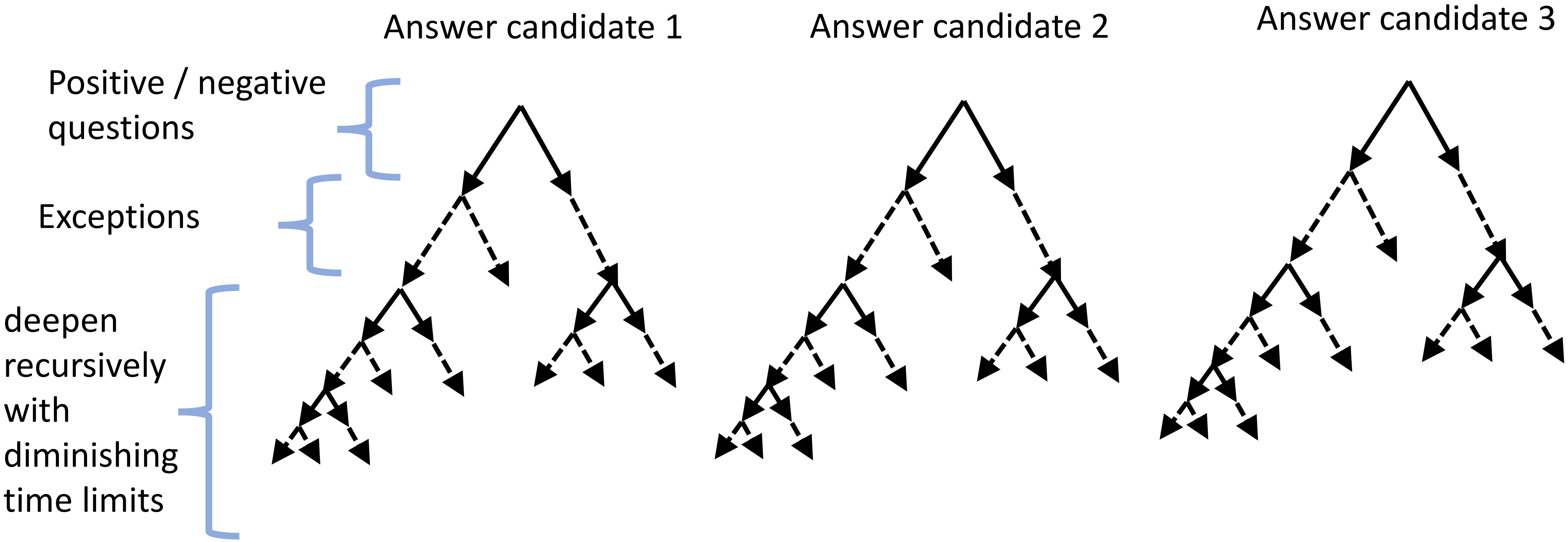
Reasoning using the GK solver



GK is built on top of our conventional high-performance resolution-based reasoner GKC, adding:

- The answer clause mechanism for finding a number of different answers.
- Finding expected proofs even if a knowledge base is inconsistent.
- Searching for both a proof of the question and a negation of the question / negation of each concrete answer.
- Estimating the numeric confidence in the statements derived from knowledge bases containing uncertain contrary and supporting evidence obtained from different sources.
- Handling exceptions by implementing default logic via recursively deepening iterations of searches with diminishing time limits.
- In progress: performing reasoning by analogy via employing known similarity scores of words along with exceptions.

Reasoning using the GK solver



Answers and Explanations in NL

If an animal likes honey, then it is probably a bear. Most bears are big, although young bears are not big. John is an animal who likes honey. Mike is a young bear. Who is big?

```
{result: answer found,

 answers:[

 {
  answer: [[$ans,some_bear]],
  blockers:
        [[$block,[$,bear,1],
         [$not,
           [prop,big,some_bear,$generic,$generic,[$ctxt,Pres,1]]]]]],
  confidence:0.85,
  positive proof:
   [
   ...,
   [7,[mp,[5,1],6,fromgoal,0.85],
     [ [$block,[$,bear,1],
       [$not,
          [prop,big,some_bear,$generic,$generic,[$ctxt,Pres,1]]]]],
      [$ans,some_bear]]]
   ]
},
```

```
{
 answer: [[$ans,c1_John]],
 blockers:
    [[$block,[$,bear,1],[$not,[prop,big,c1_John,$generic,$generic,[$ctxt,Pres,1]]]]],
      [$block,[$,animal,3],[$not,[isa,bear,c1_John]]]],
 confidence:0.765,
 positive proof:
   [
    [1,[in,frm_10,axiom,0.85],
       [[$block,[$,bear,1],[$not,[prop,big,?:X,$generic,$generic,[$ctxt,Pres,1]]]]],
        [prop,big,?:X,$generic,$generic,[$ctxt,Pres,1]],  [-isa,bear,?:X]]],
     ...
    [18,[mp,[1,2],[17,1],fromaxiom,0.765],
      [[$block,[$,bear,1],[$not,[prop,big,c1_John,$generic,$generic,[$ctxt,Pres,1]]]]],
       [$block,[$,animal,3],[$not,[isa,bear,c1_John]]]],
       [prop,big,c1_John,$generic,$generic,[$ctxt,Pres,1]]]],
 ...
    [22,[mp,[20,2],21,fromgoal,0.765],
      [[$block,[$,bear,1],[$not,[prop,big,c1_John,$generic,$generic,[$ctxt,Pres,1]]]]],
       [$block,[$,animal,3],[$not,[isa,bear,c1_John]]]],
       [$ans,c1_John]]]
 ]}
 ]}
```

Answers and Explanations in NL

If an animal likes honey, then it is probably a bear. Most bears are big, although young bears are not big. John is an animal who likes honey. Mike is a young bear. Who is big?

Likely john:

Confidence 76%.

Sentences used:

(1) If an animal likes honey, then it is probably a bear.

(2) Most bears are big, although young bears are not big.

(3) John is an animal who likes honey.

(4) Who is big?

Statements inferred:

(1) If X is a bear, then X is big. Confidence 85%. Why: sentence 2.

(2) If X does like Y and Y is a honey and X is an animal, then X is a bear. Confidence 90%. Why: sentence 1.

(4) If John has a property def1, then John does like cs4. Why: sentence 3.

...

(18) John is big. Confidence 76%. Why: statements 1, 17.

...

(21) If X matches the query, then X is an answer. Why: the question.

(22) John is an answer. Confidence 76%. Why: statements 20, 21.

Integrating LLMs?

Case study using LLMs: semantic parsing

Main alternatives:

- Learning or fine-tuning LLMs to produce a symbolic representation from the NL input.
- Prompting LLMs - without specialized fine-tuning - to produce a symbolic representation from the natural language input.
- Prompting LLMs - without specialized fine-tuning - to generate a simplified form of the input text, which is then converted to a symbolic representation by a specialized parser.

Example: GPT simplifying Wikipedia

- Prompt: Simplify, maximally shorten and split the sentence after colon to shortest possible separate subsentences,to make it understandable for children. Replace pronouns like 'they','it','he','she' in the result with nouns and proper nouns present in the result, like 'Birds can fly. Birds have feathers.' instead of 'Birds can fly. They have feathers.’
- Wikipedia sentence given after prompt: Penguins are a group of aquatic flightless birds from the order Sphenisciformes of the family Spheniscidae.
- GPT gives sentences we can parse: Penguins are special birds. They live in water. Penguins cannot fly. They are from the Sphenisciformes order. They belong to the Spheniscidae family.

Some lessons learned

The polarities+confidences+exceptions mechanism on top of FOL reasoning appears to work OK for the commonsense reasoning task.

All the main components are doable, but blue ones require a number of ML and LLM-based integrations to achieve practically usable quality:

- Semantic parsing
- Improving existing knowledge bases of “common sense”
- Wrapping and softening classical first order reasoning
- Finding logic-based proofs